

- [128] G. Lu and S. K. Debray, “Weaknesses in defenses against web-borne malware - (short paper),” in *Detection of Intrusions and Malware, and Vulnerability Assessment - 10th International Conference, DIMVA. Proceedings* (K. Rieck, P. Stewin, and J. Seifert, eds.), Lecture Notes in Computer Science, 2013.
- [129] M. Payer, “Embracing the new threat: Towards automatically self-diversifying malware,” 2014.
- [130] N. Loose, F. Mächtle, C. Pott, V. Bezsmertnyi, and T. Eisenbarth, “Madvex: Instrumentation-based Adversarial Attacks on Machine Learning Malware Detection,” *arXiv e-prints*, p. arXiv:2305.02559, May 2023.
- [131] R. Sasnauskas, Y. Chen, P. Collingbourne, J. Ketema, G. Lup, J. Taneja, and J. Regehr, “Souper: A Synthesizing Superoptimizer,” *arXiv preprint 1711.04422*, 2017.
- [132] B. G. Ryder, “Constructing the call graph of a program,” *IEEE Transactions on Software Engineering*, no. 3, pp. 216–226, 1979.
- [133] S. Narayan, C. Disselkoen, D. Moghimi, S. Cauligi, E. Johnson, Z. Gang, A. Vahldiek-Oberwagner, R. Sahita, H. Shacham, D. Tullsen, *et al.*, “Swivel: Hardening webassembly against spectre,” in *USENIX Security Symposium*, 2021.
- [134] E. Johnson, D. Thien, Y. Alhessi, S. Narayan, F. Brown, S. Lerner, T. McMullen, S. Savage, and D. Stefan, “Sfi safety for native-compiled wasm,” *NDSS. Internet Society*, 2021.
- [135] M. Willsey, C. Nandi, Y. R. Wang, O. Flatt, Z. Tatlock, and P. Panchekha, “Egg: Fast and extensible equality saturation,” *Proc. ACM Program. Lang.*, vol. 5, jan 2021.
- [136] “Stop a wasm compiler bug before it becomes a problem | fastly.” <https://www.fastly.com/blog/defense-in-depth-stopping-a-wasm-compiler-bug-before-it-became-a-problem>, 2021.
- [137] D. Cao, R. Kunkel, C. Nandi, M. Willsey, Z. Tatlock, and N. Polikarpova, “Babble: Learning better abstractions with e-graphs and anti-unification,” *Proc. ACM Program. Lang.*, vol. 7, jan 2023.
- [138] R. Tate, M. Stepp, Z. Tatlock, and S. Lerner, “Equality saturation: A new approach to optimization,” in *Proceedings of the 36th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL ’09*, (New York, NY, USA), p. 264–276, Association for Computing Machinery, 2009.
- [139] T. D. Morgan and J. W. Morgan, “Web timing attacks made practical,” *Black Hat*, 2015.

- [140] T. Schnitzler, K. Kohls, E. Bitsikas, and C. Pöpper, “Hope of delivery: Extracting user locations from mobile instant messengers,” in *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*, The Internet Society, 2023.
- [141] Mozilla, “Protections Against Fingerprinting and Cryptocurrency Mining Available in Firefox Nightly and Beta ,” 2019.
- [142] F. Cohen, “Computer viruses: theory and experiments,” *Computers & security*, vol. 6, no. 1, pp. 22–35, 1987.
- [143] P. Kocher, J. Horn, A. Fogh, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, M. Schwarz, and Y. Yarom, “Spectre attacks: Exploiting speculative execution,” in *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 1–19, 2019.
- [144] M. Schwarz, C. Maurice, D. Gruss, and S. Mangard, “Fantastic timers and where to find them: High-resolution microarchitectural attacks in javascript,” in *Financial Cryptography and Data Security* (A. Kiayias, ed.), (Cham), pp. 247–267, Springer International Publishing, 2017.
- [145] G. J. Duck, X. Gao, and A. Roychoudhury, “Binary rewriting without control flow recovery,” in *Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2020*, (New York, NY, USA), p. 151–163, Association for Computing Machinery, 2020.
- [146] S. Srikant, S. Liu, T. Mitrovska, S. Chang, Q. Fan, G. Zhang, and U.-M. O’Reilly, “Generating Adversarial Computer Programs using Optimized Obfuscations,” *arXiv e-prints*, p. arXiv:2103.11882, Mar. 2021.
- [147] H. Ye, M. Martinez, X. Luo, T. Zhang, and M. Monperrus, “Selfapr: Self-supervised program repair with test execution diagnostics,” in *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering, ASE ’22*, (New York, NY, USA), Association for Computing Machinery, 2023.
- [148] W. Zhang, S. Guo, H. Zhang, Y. Sui, Y. Xue, and Y. Xu, “Challenging machine learning-based clone detectors via semantic-preserving code transformations,” *IEEE Trans. Softw. Eng.*, vol. 49, p. 3052–3070, may 2023.

**Part II**

**Included papers**



# SUPEROPTIMIZATION OF WEBASSEMBLY BYTECODE

---

**Javier Cabrera-Arteaga**, Shrinish Donde, Jian Gu, Orestis Floros, Lucas Satabin, Benoit Baudry, Martin Monperrus

*Conference Companion of the 4th International Conference on Art, Science, and Engineering of Programming (Programming 2021), MoreVMs*

<https://doi.org/10.1145/3397537.3397567>

# CROW: CODE DIVERSIFICATION FOR WEBASSEMBLY

---

**Javier Cabrera-Arteaga**, Orestis Floros, Oscar Vera-Pérez, Benoit Baudry,  
Martin Monperrus

*Network and Distributed System Security Symposium (NDSS 2021), MADWeb*

<https://doi.org/10.14722/madweb.2021.23004>

# MULTI-VARIANT EXECUTION AT THE EDGE

---

**Javier Cabrera-Arteaga**, Pierre Laperdrix, Martin Monperrus, Benoit Baudry  
*Conference on Computer and Communications Security (CCS 2022), Moving  
Target Defense (MTD)*

<https://dl.acm.org/doi/abs/10.1145/3560828.3564007>

# WEBASSEMBLY      DIVERSIFICATION FOR MALWARE EVASION

---

**Javier Cabrera-Arteaga**, Tim Toady, Martin Monperrus, Benoit Baudry  
*Computers & Security, Volume 131, 2023*

<https://www.sciencedirect.com/science/article/pii/S0167404823002067>



# WASM-MUTATE: FAST AND EFFECTIVE BINARY DIVERSIFICATION FOR WEBASSEMBLY

---

**Javier Cabrera-Arteaga**, Nick Fitzgerald, Martin Monperrus, Benoit Baudry  
*Under revision*

# SCALABLE COMPARISON OF JAVASCRIPT V8 BYTECODE TRACES

---

**Javier Cabrera-Arteaga**, Martin Monperrus, Benoit Baudry

*11th ACM SIGPLAN International Workshop on Virtual Machines and  
Intermediate Languages (SPLASH 2019)*

<https://doi.org/10.1145/3358504.3361228>